Omerta: A Trust-Based Alternative to Blockchain Consensus for Decentralized Compute Markets

Participation Verification Through Subjective Trust and Automated Monetary Policy

Technical Whitepaper

January 2026

Contents

Abstract

Decentralized compute sharing faces a fundamental challenge: how can strangers cooperate without a trusted central authority? Blockchain-based systems address this through proof-of-work or proof-of-stake consensus, achieving Byzantine fault tolerance at the cost of significant resource overhead and limited throughput. We present Omerta, an alternative architecture that sidesteps Byzantine consensus entirely by embracing subjective trust computed locally from verifiable on-chain data.

The key insight underlying Omerta is that compute markets do not require global agreement—a consumer only needs to know whether a specific provider will deliver to *them*. By computing trust relative to the observer's position in the transaction graph, Omerta limits the impact of reputation attacks and enables natural scaling without the coordination overhead of global consensus.

This paper presents the complete Omerta system design, including: (1) a trust accumulation model based on verified transactions rather than subjective ratings; (2) local trust computation that prevents cross-community trust arbitrage; (3) an on-chain order book for price discovery; (4) identity-bound VM access that makes credential theft useless; (5) detection mechanisms for Sybil clusters, collusion rings, and trust manipulation; and (6) automated monetary policy that adjusts economic parameters in response to detected threats.

We argue that the claimed "trustlessness" of blockchain systems is largely illusory—social consensus underlies all distributed systems and emerges when sufficient value is at stake. Omerta makes this social layer explicit, providing transparent mechanisms to track and verify trust relationships. The result is a system that mirrors how human trust networks actually function: imperfect rules made workable through aligned incentives and gradual reputation building.

1. Introduction

The vision of decentralized computing—where anyone can contribute resources and anyone can consume them, without intermediaries extracting rents—has motivated decades of research. From early peer-to-peer file sharing networks through blockchain-based systems to contemporary distributed computing initiatives, the core challenge remains: how do strangers cooperate when each has incentive to defect?

Traditional approaches fall into two categories. **Trusted intermediary** models, exemplified by cloud computing providers, centralize authority in organizations that can enforce contracts and punish misbehavior. This works but introduces single points of failure, censorship risk, and rent extraction. **Blockchain-based** models, exemplified by Ethereum and its descendants, replace trusted intermediaries with cryptographic consensus protocols that theoretically eliminate the need for trust. This also works but imposes significant costs: massive energy expenditure (proof-of-work), capital lockup requirements (proof-of-stake), limited transaction throughput, and delayed finality.

We propose a third path. Omerta is a trust-based distributed compute network that neither centralizes authority nor attempts to eliminate trust. Instead, it makes trust *subjective*, *local*, and *earned*—computed by each participant from their own position in the network, based on verifiable on-chain data accumulated over time.

1.1 The Trust Spectrum

Trustlessness is not binary—it exists on a spectrum. Proof-of-work and proof-of-stake mechanisms genuinely increase trustlessness compared to centralized alternatives. They represent real achievements in distributed systems research. However, historical episodes demonstrate that no system achieves absolute trustlessness:

The DAO Hack (2016): An attacker exploited a smart contract vulnerability to drain $60 million from The DAO. The Ethereum community responded with a hard fork that reversed the theft—creating Ethereum (rolled back) and Ethereum Classic (preserved the "immutable" history). The community chose social consensus over mechanical execution.

Bitcoin Value Overflow (2010): A bug created 184 billion bitcoins out of thin air. Developers and node operators coordinated to deploy a fix and roll back the chain. Human judgment overrode the protocol when stakes were high enough.

Exchange Coordination: When $40 million was stolen from Binance in 2019, the company seriously considered coordinating a Bitcoin rollback before deciding against it. The option existed—revealing that social coordination remains available when needed.

These episodes do not invalidate blockchain achievements. Rather, they reveal that we operate on a spectrum from full trust (centralized authority) to reduced trust (cryptographic consensus) to some irreducible social layer that emerges under sufficient pressure. No practical system reaches the zero-trust endpoint.

The question becomes: given that we cannot achieve absolute trustlessness anyway, what are we paying for the trustlessness we do achieve? And could we relax our requirements slightly to capture most of the practical benefit at dramatically lower cost?

This is the same reasoning that motivates ephemeral compute over fully homomorphic encryption (FHE). FHE provides the ultimate guarantee: compute on encrypted data without ever decrypting

it. No trust in the compute provider required. But FHE imposes 1,000-1,000,000x computational overhead [21], making it impractical for most workloads. Ephemeral compute—where data exists briefly on untrusted hardware with verification and economic penalties—provides weaker guarantees but serves far more use cases at practical cost.

Omerta applies this spectrum thinking to consensus itself. Blockchain consensus mechanisms genuinely reduce trust requirements, but at significant cost: energy expenditure (PoW), capital lockup (PoS), limited throughput, and delayed finality. We ask: for compute markets specifically, can we relax the global consensus requirement while preserving the practical security properties that matter?

Our hypothesis is yes. Compute markets do not require global agreement—they require pairwise trust between specific buyers and sellers. By computing trust locally rather than achieving global consensus, Omerta aims to capture most of the practical benefit of decentralization at dramatically lower cost, making trustworthy compute sharing accessible to more people.

1.2 Our Contribution

This paper makes the following contributions:

1. **A subjective trust model** where each participant computes trust locally based on their own transaction history and the assessments of parties they themselves trust, rather than relying on global consensus.
2. **Local trust computation** that propagates trust through the transaction graph with decay, limiting the impact of attacks that exploit trust earned in distant communities.
3. **Verified trust accumulation** where trust derives from on-chain transaction records rather than subjective ratings, eliminating the attack surface of fake feedback.
4. **Age as unforgeable credential**: Identity age—time since creation—is the one credential that cannot be purchased or manufactured, providing a temporal Sybil defense.
5. **Automated monetary policy** that adjusts economic parameters (payment curves, transfer burns, detection thresholds) in response to observed network metrics.
6. **Explicit security analysis** distinguishing protections that must be absolute (UBI distribution, trust from activity) from those where some adversarial advantage may be tolerated (risk diversification, community separation).

The remainder of this paper is organized as follows. Section 2 reviews related work on reputation systems, Sybil resistance, and blockchain consensus. Section 3 presents the system architecture. Section 4 details the trust model. Section 5 describes the economic mechanisms. Section 6 analyzes attack vectors and defenses. Section 7 presents simulation results. Section 8 discusses limitations and future work. Section 9 concludes.

2. Related Work

2.1 Reputation Systems

The challenge of establishing trust among strangers online has motivated extensive research on reputation systems. Resnick et al. [1] identified the core requirements: long-lived identities, captured feedback, and feedback-guided decisions. The eBay feedback mechanism demonstrated these principles at scale, though its binary ratings created opportunities for manipulation [2].

More sophisticated approaches emerged from peer-to-peer networks. **EigenTrust** [3] computed global trust through iterative aggregation similar to PageRank, but remained vulnerable to strategic manipulation by colluding peers. **PeerTrust** [4] incorporated transaction context but required honest reporting. **Credence** [5] enabled subjective reputation evaluation but focused on content authenticity rather than service quality.

Omerta builds on these foundations while addressing key limitations. Unlike EigenTrust's global scores, Omerta computes trust relative to the observer. Unlike systems treating ratings as exogenous, Omerta derives trust from transaction records. Unlike systems assuming honest reporting, Omerta makes trust computable from verifiable on-chain facts.

2.2 Sybil Resistance

Douceur [6] proved that without a trusted central authority, a single adversary can present arbitrarily many identities indistinguishable from honest participants. This "Sybil attack" undermines any reputation system where influence scales with identity count.

Defenses fall into three categories:

Resource-based: Require each identity to demonstrate control of scarce resources—computational power [7], financial stake [8], or hardware attestation. Effective but expensive.

Social-based: Leverage trust graph structure, noting that Sybil identities have sparse connections to honest nodes [9, 10]. Effective when social graph is meaningful.

Temporal: Require identities to exist over time before gaining influence. Attackers can still create identities in advance, but cannot accelerate trust accumulation.

Omerta employs a hybrid approach: economic penalties (transfer burns), social detection (cluster analysis), and most importantly, temporal constraints. Identity age—time since on-chain creation—cannot be forged, purchased, or accelerated. This makes Sybil attacks expensive in the dimension attackers cannot optimize: time.

2.3 Blockchain Consensus

Bitcoin [7] introduced proof-of-work consensus, achieving Byzantine fault tolerance through computational cost. Subsequent systems explored alternatives: proof-of-stake [8], delegated proof-of-stake [11], practical Byzantine fault tolerance [12], and various hybrid approaches.

All these mechanisms solve the Byzantine Generals Problem: achieving agreement among distributed parties despite malicious actors. This requires $n \geq 3f + 1$ nodes to tolerate f failures, with significant coordination overhead.

Omerta sidesteps this problem entirely. Compute markets do not require global consensus—they require pairwise trust between specific buyers and sellers. By computing trust locally, Omerta eliminates the coordination overhead of global agreement while providing the security properties actually needed for compute rental.

2.4 Secure Computation Approaches

The ultimate solution to untrusted compute would be **fully homomorphic encryption (FHE)** [21], which enables computation on encrypted data without decryption. FHE provides mathematical guarantees: the compute provider learns nothing about the data. However, current FHE implementations impose 1,000-1,000,000x overhead compared to plaintext computation [22], restricting practical use to narrow applications.

Trusted execution environments (TEEs) like Intel SGX [23] provide hardware-based isolation, but require trusting the hardware manufacturer and have been vulnerable to side-channel attacks [24]. **Secure multi-party computation (MPC)** [25] distributes computation across parties such that no single party learns the inputs, but requires coordination and communication overhead scaling with circuit complexity.

These approaches represent one end of the trust spectrum: maximum guarantees at maximum cost. Omerta explores the opposite trade-off: accepting some trust requirements in exchange for practical performance that serves more use cases.

2.5 Decentralized Computing

Distributed computing projects like BOINC [13], Folding@home [14], and SETI@home demonstrated that volunteers would contribute compute resources for scientific research. Commercial successors like Golem [15] and iExec [16] built on blockchain platforms to enable paid compute sharing.

These systems face common challenges: verifying that claimed work was actually performed, preventing providers from delivering inferior resources, and detecting collusion. Omerta addresses these through continuous verification, trust-based payment splits, and statistical detection of manipulation patterns.

3. System Architecture

3.1 Design Principles

Omerta is built on several core principles:

Verifiable Facts, Subjective Trust: The blockchain stores facts (transactions, verification logs, assertions). Trust scores are computed locally by each participant from these facts according to their own criteria.

Identity Age as Credential: Of all possible credentials, only time-on-chain cannot be manufactured. New identities start with nothing and must earn trust through participation.

Uniform Pricing, Trust-Based Splits: All consumers pay the same market rate. Trust determines how payments split between provider and burn. High trust = more to provider. Low trust = more burned.

Earned Write Permission: Only identities above trust thresholds can write to the chain, with conflict resolution through trust-weighted voting. Trust score IS the stake.

3.2 On-Chain Data

The blockchain records five types of data:

Identity Records: Public key, creation timestamp, and signature capabilities. All derived data (age, transaction count, trust scores) is computed from other records.

Transactions: Consumer and provider identities, amounts paid/received/burned, resource specifications, duration, and signatures from both parties.

Verification Logs: Results of resource checks, including verifier identity, claimed vs. measured resources, pass/fail result, and verifier signature.

Trust Assertions: Signed claims by one identity about another, including score, classification (positive or negative), evidence hashes, and reasoning.

Order Book: Bids and asks for compute resources, enabling price discovery through market mechanisms.

3.3 Market Structure

Omerta implements an on-chain order book for price discovery:

Order Types: Bids (consumer wants to buy) and asks (provider wants to sell), with resource specifications, price, duration constraints, and expiration.

Resource Classes: Standardized categories (small_cpu, medium_cpu, gpu_consumer, gpu_datacenter) enabling liquidity aggregation.

Matching Engine: Price-time priority matching. When orders cross, sessions initiate and escrow triggers.

Spot Rate: Volume-weighted average of recent trades, providing simple consumer-facing pricing while preserving price discovery.

Consumers see simple pricing ("8 cores = 0.09 OMC/hr") without needing to understand the underlying market mechanics.

3.4 Session Lifecycle

A compute session proceeds as follows:

1. **Order Placement**: Consumer places bid or provider places ask
2. **Matching**: Orders cross, session initiates
3. **Escrow Lock**: Consumer's payment locked in escrow
4. **VM Configuration**: Provider configures VM with consumer's identity key for access
5. **Compute Execution**: Consumer uses resources
6. **Verification**: Random audits check resource claims
7. **Settlement**: Escrow released based on outcome and trust scores

Either party can terminate at any time—the market handles quality through consumer exit and provider reliability signals.

4. Trust Model

4.1 Trust Accumulation

Trust accumulates from verified transactions, not subjective ratings:

$$T_{base} = T_{transactions} + T_{assertions}$$

$$T_{transactions} = \sum_i (CREDIT \times resource\_weight_i \times duration_i \times verification\_score_i \times cluster\_weight_i)$$

Transaction-based trust grows with verified compute provision. Each term serves a specific purpose: resource weights normalize across compute types, duration captures commitment extent, verification scores reflect audit outcomes, and cluster weights downweight suspected Sybil transactions.

Assertion-based trust adjusts for reported incidents:

$$T_{assertions} = \sum_i (score_i \times credibility_i \times decay_i)$$

Assertions are signed reports of specific incidents with scores in [-1, 1]. Positive scores (commendations) add trust; negative scores (violations) subtract. Credibility derives from the asserter's own trust, creating recursive dependency resolved through iterative computation.

4.2 Age as Derate Factor

A critical design choice: age should **never add trust**, only remove a penalty from young identities:

$$T_{effective} = T_{base} \times age\_derate$$

$$age\_derate = \min\left(1.0, \frac{identity\_age}{AGE\_MATURITY\_DAYS}\right)$$

New identities start at zero effective trust regardless of transaction volume. This prevents attackers from pre-creating dormant identities that accumulate trust through mere existence. You can only earn trust by participating over time.

4.3 Local Trust Computation

Trust is not global. Each observer computes trust relative to their position in the network:

$$T(subject, observer) = T_{direct} + T_{transitive}$$

$$T_{transitive} = \sum_{intermediary} T(intermediary, observer) \times T(subject, intermediary) \times DECAY^{path\_length}$$

Direct trust comes from personal transaction history. Transitive trust propagates through trusted intermediaries with exponential decay per hop.

Why this matters: An attacker cannot build trust in Community A and exploit it in Community B. Observers in B see the attacker's trust discounted by lack of network path. The attacker must build trust directly with each community they wish to exploit—exactly how human trust works.

4.4 Parameterized Infractions

Not all violations are equal. Infraction severity scales with potential network impact:

$$effective\_score = base\_score \times impact\_multiplier \times context\_multiplier$$

Impact scales with transaction value, resources affected, and duration. Context includes repeat offense history. This creates appropriate gray areas—small mistakes don't destroy trust, while large attacks risk proportional penalties.

5. Economic Mechanisms

5.1 Payment Splits

Consumer payments split between provider and cryptographic burn based on provider trust:

$$provider\_share = 1 - \frac{1}{1 + K_{PAYMENT} \times T}$$

As trust increases, provider share approaches 100% asymptotically but never reaches it. New providers with low trust see most of their payment burned; established providers keep most of it.

This creates natural incentives: build trust to keep more of your earnings. No external enforcement needed—the economics handle it.

5.2 Transfer Burns

Transfers between identities are taxed based on the minimum trust of sender and receiver:

$$transfer\_burn\_rate = \frac{1}{1 + K_{TRANSFER} \times \min(T_{sender}, T_{receiver})}$$

Low-trust identities cannot easily transfer coins. This prevents reputation laundering (build trust, exploit, transfer coins to fresh identity) and creates strong incentive to donate compute rather than buy coins.

5.3 Daily Distribution

New coins are minted daily and distributed proportionally to trust scores:

$$daily\_share(i) = \frac{effective\_trust(i)}{\sum_j effective\_trust(j)} \times DAILY\_MINT$$

This creates the core incentive: misbehave and your trust drops, reducing tomorrow's share. The motivation to be honest is simply: keep getting your handout.

5.4 Donation and Negative Bids

Providers can accept negative-price bids from research organizations, burning their own coins for accelerated trust:

Bid Type	Price	Trust Multiplier
Commercial	Positive	1x
Zero donation	Zero	1x
Negative donation	Negative	Up to 4x

This enables bootstrapping: new providers can burn coins to accelerate trust building while providing verified compute to research projects. Trust cannot be purchased without actual work—the burn is additional, not replacement.

6. Attack Analysis and Defenses

6.1 Sybil Attacks

Attack: Create many fake identities to manipulate trust or distribution.

Defenses: - **Age derate**: New identities earn nothing initially - **Cluster detection**: Tightly-connected subgraphs with few external edges are flagged - **Behavioral similarity**: Identities behaving too similarly are downweighted - **Activity requirements**: Must maintain ongoing participation

6.2 Collusion and Trust Inflation

Attack: Colluding parties mutually vouch for each other to inflate trust.

Defenses: - **Graph analysis**: Detect circular trust flows and isolated cliques - **Verification requirements**: Trust requires verified compute, not just assertions - **Statistical anomaly detection**: Burst activity, coordinated timing flagged

6.3 Trust Arbitrage

Attack: Build trust in Community A, exploit in Community B.

Defense: Local trust computation means trust doesn't transfer across network distance. Attackers must build trust directly with each target community.

6.4 Multi-Identity Exploitation

Attack: Maintain multiple identities to hedge risk or enable sacrificial attacks.

Critical Distinction: Some multi-identity strategies must be absolutely prevented; others may be tolerated.

Absolute Protections: - UBI distribution: Malicious behavior in any linked identity reduces combined distribution - Trust from activity: Same work split across N identities yields at most single-identity trust - Accusation credibility: N low-credibility accusations don't sum to high credibility

Tolerated Advantages: - Risk diversification: Legitimate businesses may operate multiple identities - Community separation: Operating in isolated communities without cross-contamination - Recovery via new identity: Starting fresh with appropriate penalties

6.5 Identity-Bound Access

Problem: Traditional credential theft enables exploiting stolen identity's reputation.

Solution: VM access is bound to the consumer's on-chain private key. No key, no access. If you have the private key, you ARE the identity—there is no "stealing," only "being."

7. Simulation Results

We conducted simulation studies testing the automated monetary policy under adversarial conditions. Key findings:

7.1 Attack Scenario Outcomes

Scenario	Final Gini	Cluster Prevalence	Policy Response
Baseline (Honest)	0.783	0.000	Stable
Trust Inflation	0.706	0.250	K_TRANSFER increased
Sybil Explosion	0.641	0.545	ISOLATION_THRESHOLD decreased
Gini Manipulation	0.882	0.000	K_PAYMENT decreased

7.2 Key Finding: Structural vs. Parametric Attacks

The simulations revealed a striking pattern: automated policy adjustments trigger correctly but have limited impact on final outcomes. This suggests that attack effects are primarily structural rather than parameter-dependent.

Implication: Effective attack resistance requires architectural defenses that make attacks structurally infeasible, complemented by policy adjustments for fine-tuning. Parameter tweaking alone is insufficient against determined adversaries.

7.3 Long-Term Stability

Five-year simulations showed stable trust accumulation under honest conditions and recovery between attack waves under adversarial conditions. The core trust mechanisms prove robust to extended adversarial pressure.

8. Discussion

8.1 The Trust-Cost Spectrum

Different approaches occupy different positions on the trust-cost spectrum:

Approach	Trust Required	Cost	Practical Scope
Centralized cloud	High (trust provider)	Low	Broad
FHE	None	Extreme (1000x+)	Very narrow
TEE (SGX)	Medium (trust hardware)	Low-Medium	Medium
PoW blockchain	Low	High (energy)	Medium
PoS blockchain	Low-Medium	Medium (capital)	Medium
Omerta	Medium (trust earned)	Low	Broad

The key insight is that blockchain approaches—while genuinely reducing trust requirements—may not occupy the optimal point for compute markets. They pay significant costs (energy, capital, throughput limits) for global consensus that compute markets may not need.

8.2 What Blockchain Achieves

We should be precise about what blockchain consensus mechanisms accomplish:

Genuine achievements: - Coordination without designated coordinator - Resistance to unilateral censorship - Auditable history without trusted record-keeper - Credible monetary policy without central bank

These are real and valuable. PoW and PoS represent genuine advances in reducing trust requirements.

What the historical episodes reveal: - The spectrum has no zero-trust endpoint in practice - Social consensus remains available when stakes are high enough - This doesn't invalidate the achievements—it bounds them

8.3 Omerta's Position

Omerta bets that for compute markets specifically, we can relax global consensus while preserving the properties that matter:

Property	Blockchain Approach	Omerta Approach	Trade-off
Double-spend prevention	Global UTXO consensus	Escrow locks before session	Equivalent for sessions
History integrity	PoW/PoS difficulty	Trust-weighted writes	Weaker globally, sufficient locally
Sybil resistance	Computational/capital cost	Time cost (age)	Different attack economics

Property	Blockchain Approach	Omerta Approach	Trade-off
Censorship resistance	Anyone can mine/stake	Anyone above trust threshold	Requires earning entry

The hypothesis is that these trade-offs are acceptable for compute markets, where: - Transactions are bilateral (buyer-seller), not global transfers - Sessions are ephemeral, not permanent state - Verification is possible during execution - Reputation has natural meaning (did you deliver?)

8.4 Analogy to FHE vs. Ephemeral Compute

The relationship between Omerta and blockchain mirrors the relationship between ephemeral compute and FHE:

	Maximum Guarantee	Practical Alternative
Data privacy	FHE (compute on encrypted)	Ephemeral compute (brief exposure + verification)
Consensus	Global Byzantine (PoW/PoS)	Local trust (Omerta)
Cost	1000x+ overhead	Near-native performance
Accessibility	Narrow applications	Broad applicability

In both cases, relaxing the maximum guarantee enables serving far more users at practical cost. The question is whether the relaxed guarantee is sufficient for the use case.

8.5 Making the Social Layer Explicit

Every distributed system ultimately has a social layer. Bitcoin's ledger has been modified by human coordination. Ethereum's code yielded to community override. The question is not whether humans are trusted, but which humans and how.

Omerta makes this explicit rather than hiding it beneath claims of mathematical trustlessness. Trust relationships are tracked on-chain. Incentives align through economics. Bad actors are identified over time. This mirrors how working human institutions operate—imperfect rules made workable through aligned incentives.

8.6 Interoperability Across the Spectrum

Different points on the trust-cost spectrum suit different use cases. A mature ecosystem might include multiple networks that interoperate, with value and workloads flowing to appropriate trust levels.

Use cases by trust level:

Trust Level	Example Use Cases	Why This Level
Maximum (FHE/MPC)	Medical records analysis, financial audits, voting	Data must never be exposed, even briefly

Trust Level	Example Use Cases	Why This Level
High (PoW/PoS blockchain)	Digital currency, smart contracts with large stakes, cross-border settlements	Global consensus needed, high-value permanent state
Medium (Omerta-style)	General compute rental, batch processing, development environments, CI/CD	Bilateral transactions, ephemeral sessions, verification possible
Lower (reputation only)	Content delivery, caching, non-sensitive workloads	Speed matters more than guarantees, easy to verify after the fact

Cross-network flows:

Networks at different trust levels can bridge to each other:

- **Settlement on high-trust chains**: An Omerta-style network could settle periodic summaries to a PoS blockchain, gaining the permanence guarantees of global consensus for aggregate state while handling high-frequency bilateral transactions locally.
- **Escalation for disputes**: Normal compute sessions run on medium-trust infrastructure. Disputed sessions escalate to higher-trust arbitration—perhaps a smart contract on Ethereum that evaluates cryptographic evidence.
- **Sensitive workload isolation**: A pipeline might run preprocessing on Omerta (cheap, fast), then route sensitive computation to FHE or TEE enclaves, then aggregate results back on Omerta.
- **Trust bootstrapping**: New participants could establish initial reputation on a high-trust chain (where Sybil attacks are expensive), then bridge that identity to lower-cost networks.

The vision:

Rather than one network attempting to serve all use cases at one trust level, the ecosystem stratifies. Users and workloads flow to appropriate levels based on their actual security requirements. Most computation doesn't need the guarantees (or costs) of global consensus. Some does. A well-designed ecosystem makes both available and composable.

This is analogous to how the internet layers protocols: TCP provides reliable delivery at some cost, UDP provides speed without guarantees, and applications choose based on their needs. Similarly, a trust-stratified compute ecosystem would let applications choose their trust level, paying only for the guarantees they actually require.

8.7 Limitations

Bootstrap Problem: The network requires initial trusted participants to establish the trust graph. Genesis block contents define starting conditions.

Sophisticated Attackers: Well-resourced attackers willing to invest years in building reputation before striking remain a threat. No system fully prevents long-con attacks.

Parameter Sensitivity: Optimal parameter values require empirical tuning and may vary across network conditions.

Verification Overhead: Random audits impose costs on honest participants. The verification rate must balance security against efficiency.

9. Conclusion

Trustlessness exists on a spectrum. Proof-of-work and proof-of-stake mechanisms genuinely reduce trust requirements compared to centralized alternatives—this is a real achievement. But they do so at significant cost, and historical episodes demonstrate that no practical system reaches the zero-trust endpoint.

Omerta explores a different point on this spectrum. Rather than paying for global consensus that compute markets may not need, Omerta computes trust locally based on verifiable on-chain data. The hypothesis is that for bilateral compute transactions—where sessions are ephemeral, verification is possible during execution, and reputation has natural meaning—local trust provides sufficient security at dramatically lower cost.

This parallels the choice between fully homomorphic encryption and ephemeral compute. FHE provides the ultimate guarantee but at 1000x+ overhead, restricting practical use. Ephemeral compute accepts some trust requirements in exchange for serving far more use cases. Similarly, blockchain consensus provides strong guarantees but at costs (energy, capital, throughput) that may exceed what compute markets require.

Our simulation studies validate that automated policy mechanisms respond appropriately to detected threats, while revealing that parameter adjustment alone cannot counter structural attacks. This suggests that effective systems need architectural defenses that make attacks infeasible by design, complemented by policy mechanisms for fine-tuning.

The goal is not to replace blockchain systems—they serve real purposes and represent genuine advances. The goal is to expand the design space, recognizing that different applications may have different optimal points on the trust-cost spectrum. For compute markets specifically, we believe there is an underexplored region that could make trustworthy compute sharing accessible to more people at practical cost.

Omerta is an experiment in finding that region.

References

[1] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems," *Communications of the ACM*, vol. 43, no. 12, pp. 45-48, 2000.

[2] C. Dellarocas, "The digitization of word of mouth: Promise and challenges of online feedback mechanisms," *Management Science*, vol. 49, no. 10, pp. 1407-1424, 2003.

[3] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust algorithm for reputation management in P2P networks," in *Proc. WWW*, 2003.

[4] L. Xiong and L. Liu, "PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE TKDE*, vol. 16, no. 7, pp. 843-857, 2004.

[5] K. Walsh and E. G. Sirer, "Experience with an object reputation system for peer-to-peer file-sharing," in *Proc. NSDI*, 2006.

[6] J. R. Douceur, "The Sybil attack," in *Proc. IPTPS*, 2002.

[7] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[8] S. King and S. Nadal, "PPCoin: Peer-to-peer crypto-currency with proof-of-stake," 2012.

[9] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "SybilGuard: Defending against Sybil attacks via social networks," *ACM SIGCOMM*, 2006.

[10] G. Danezis and P. Mittal, "SybilInfer: Detecting Sybil nodes using social networks," in *Proc. NDSS*, 2009.

[11] D. Larimer, "Delegated proof-of-stake (DPOS)," *Bitshare whitepaper*, 2014.

[12] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. OSDI*, 1999.

[13] D. P. Anderson, "BOINC: A system for public-resource computing and storage," in *Proc. Grid*, 2004.

[14] V. S. Pande et al., "Atomistic protein folding simulations on the submillisecond time scale using worldwide distributed computing," *Biopolymers*, vol. 68, no. 1, pp. 91-109, 2003.

[15] The Golem Project, "The Golem whitepaper," 2016.

[16] iExec, "iExec: Blockchain-based decentralized cloud computing," 2017.

[17] M. Feldman, K. Lai, I. Stoica, and J. Chuang, "Robust incentive techniques for peer-to-peer networks," in *Proc. EC*, 2004.

[18] R. Jurca and B. Faltings, "Collusion-resistant, incentive-compatible feedback payments," in *Proc. EC*, 2007.

[19] G. E. Bolton, B. Greiner, and A. Ockenfels, "Engineering trust: Reciprocity in the production of reputation information," *Management Science*, vol. 59, no. 2, pp. 265-285, 2013.

[20] D. E. Denning, "An intrusion-detection model," *IEEE TSE*, vol. 13, no. 2, pp. 222-232, 1987.

[21] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. STOC*, 2009.

[22] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?," in *Proc. CCSW*, 2011.

[23] V. Costan and S. Devadas, "Intel SGX explained," *IACR Cryptology ePrint Archive*, 2016.

[24] J. Van Bulck et al., "Foreshadow: Extracting the keys to the Intel SGX kingdom," in *Proc. USENIX Security*, 2018.

[25] A. C. Yao, "How to generate and exchange secrets," in *Proc. FOCS*, 1986.

Appendix: Key Parameters

Parameter	Description	Typical Range
K_PAYMENT	Payment curve slope	0.01 - 0.10
K_TRANSFER	Transfer burn slope	0.01 - 0.10
AGE_MATURITY_DAYS	Days to full trust potential	90 - 180
DAILY_MINT	Coins minted per day	Decreasing schedule
ISOLATION_THRESHOLD	Sybil cluster detection	0.7 - 0.9
TRANSITIVITY_DECAY	Trust decay per hop	0.5 - 0.8
DAMPENING_FACTOR	Policy adjustment scaling	0.1 - 0.5